



UNIVERSITI PUTRA MALAYSIA

**THE USAGE OF OBJECT ORIENTED APPROACH IN SOFTWARE
DEVELOPMENT COMPANIES**

ANNE VIKNESW ARY

FSKTM 2003 15

**THE USAGE OF OBJECT ORIENTED APPROACH IN SOFTWARE
DEVELOPMENT COMPANIES**

By

ANNE VIKNESWARY

Thesis submitted in fulfillment of the requirement for the Degree of Master of Science
in the Faculty of Computer Science and Information Technology

University Putra Malaysia

October 2003

Abstract of thesis presented to the Senate of University Putra Malaysia in fulfillment of
the requirement for the Degree of Master of Science

**THE USAGE OF OBJECT ORIENTED APPROACH IN SOFTWARE
DEVELOPMENT COMPANIES**

By

ANNE VIKNESWARY

October 2003

Chairperson : **Pn. Norhayati Mohd Ali**

Faculty : **Computer Science and Information Technology**

Object orientation (OO) has become a new buzzword for computer applications. Things are touted as being object oriented, with the implication that this is the “new” and “better” way to do things. As there are many approaches used to develop systems, therefore the software industry has to cope with the rapid technological evolution and the global market competition, in order to satisfy the growing user demands for quality, services and bandwidth.

This project seeks a review of the status of the object-oriented approach among the software industry. Based on this, I will analyze the awareness of the object-oriented

approach among software developers. In addition to this, the level of usage of the object-oriented approach in software development activities will also be examined. The results obtained will be statistically analyzed using SPSS (Statistical Package for Social Sciences), and discussed later in the light of the literature reviewed.

ACKNOWLEDGEMENT

I would like to take this opportunity to thank all those who had helped in one way or another in this project. Without their unending support and guidance, I might have not proceeded this far towards completing it in the given timeframe.

First and foremost, my utmost gratitude goes to God Almighty for giving me the perseverance and strength needed to work on this piece of literature and making it possible in every way.

Many thanks to my supervisor, **PN. NORHAYATI MOHD ALI** for her undivided attention and furnishing me with much guidance as and when I stumble upon any difficulties during this phase.

A heap of appreciation goes out to my parents who were always there to push and encourage me whenever I tend to lose focus during these trying times and I'm totally grateful and honored for their support and love.

Lastly, many thanks again to all my friends and colleagues for their support and encouragement for without those, the completion of this project would not have been possible.

APPROVAL

The project paper is submitted to the Faculty of Computer Science and Information Technology, UPM and was accepted as fulfillment of the requirement for the Degree of Masters of Science.

Approved by



PN. NORHAYATI MOHD ALI

Project Supervisor

Faculty of Computer Science and Information Technology

University Putra Malaysia

Date: 19 NOV 2003

APPROVAL

The project paper is submitted to the Faculty of Computer Science and Information Technology, UPM and was accepted as fulfillment of the requirement for the Degree of Masters of Science.

Approved by



PN. NORHAYATI MOHD ALI

Project Supervisor

Faculty of Computer Science and Information Technology

University Putra Malaysia

Date: 19 NOV 2003

DECLARATION

I hereby declare that this thesis, submitted to University Putra Malaysia as a partial fulfillment of the requirements for the Master of Science (Computer Science) degree has not been submitted for any other degree at UPM or other institutions. I also certify that the work described here is entirely my own except for excerpts and summaries, which have been duly acknowledged.

Signed



Candidate

Anne Vikneswary

Date: 19/11/03

TABLE OF CONTENTS

	Page
ABSTRACT	ii
ACKNOWLEDGEMENT	iv
APPROVAL	v
DECLARATION	vi
LIST OF TABLES	ix
LIST OF FIGURES	x

CHAPTER

1. INTRODUCTION

1.1	Background	1
1.2	Problem Statement	4
1.3	Objective of the project	4
1.4	Scope of the project	5

2. LITERATURE REVIEW

2.1	Background of object-oriented approach	6
2.1.1	Object-oriented Programming Languages	6
2.1.2	Object-oriented Methodologies	7
2.1.3	Object-oriented Databases	7
2.2	Impact on Introducing Object-oriented Approach	9
2.3	Benefits of Object-oriented Approach	18
2.4	Challenges to Implement Object-oriented Approach	22



3.	METHODOLOGY	
3.1	Research Methodology	24
3.1.1	Identifying the software development companies	25
3.1.2	Developing the Questionnaire	25
3.1.2.1	Why Qualitative Research?	26
3.1.3	Distributing and Collecting the Questionnaire	27
3.1.4	Analyzing the Questionnaire using SPSS 11.5	28
3.1.5	Results	28
3.2	Data Analysis Procedure	29
3.2.1	Introduction to SPSS	29
3.2.2	Overview of SPSS	30
3.2.3	Use of SPSS for Entering Data	30
3.2.4	The Output Viewer in SPSS	32
4.	RESULTS & DISCUSSION	
4.1	Variables of Analysis	34
5.	CONCLUSION	69
	REFERENCES	75
	APPENDIX A	77
	APPENDIX B	85

LIST OF TABLES

Table	Description	Page
2.1	CASE Tools and Development Environment	10
2.2	Metrics Comparison between Existing System and Object-oriented System	12
2.3	Comparison of Time Spent in Each Phase Between Object-oriented and Existing Methodology	14
3.1	Actual Number of Responses	27
5.1	Areas of Object-oriented Approach that Users are Familiar	69
5.2	Size of Systems Developed	71
5.3	Approach Used for System Development	71
5.4	Software Development Approach Used	72
5.5	Phases of Software Development that Use the Object-oriented Approach	73
5.6	The Benefits of Object-oriented Approach	74
5.7	Problems Faced in using Object-oriented Approach	74

LIST OF FIGURES

Figure	Description	Page
2.1	Roles and Training Requirement for SCOOP	13
2.2	Cost over Time Graph	17
3.1	The Steps for the Project	24
3.2	Data Editor for SPSS 11.5	31
3.3	The Output Viewer	32
4.1	Graph – Respondents Position (%)	35
4.2 (a)	Graph – Working Experience (%)	36
4.2 (b)	Graph – Working Experience vs. Position (%)	37
4.3	Graph – Types of Approaches (%)	38
4.4 (a)	Graph – Number of Staff Involved in Software Development (%)	39
4.4 (b)	Graph – Number of Staff Involved in Software Development vs. Position	40
4.5	Graph – Familiarity of Object-oriented Approach (%)	42
4.6	Graph – Ways of getting to know the Object-oriented Approach (%)	43
4.7	Graph – Familiar Areas of the Object-oriented Approach (%)	44
4.8	Graph – Familiar Programming Languages (%)	45
4.9	Graph – Familiar Object-oriented Analysis and Design (OOAD) Methodologies (%)	47
4.10	Graph – Familiar Object-oriented Databases (%)	48

4.11	Graph – Approach Used for System Development (%)	49
4.12	Graph – Rating of Experience Level (%)	51
4.13	Graph – Frequent Involvement (%)	52
4.14	Graph – Number of years using the Object-oriented Approach (%)	54
4.15	Graph – Size of Systems Developed using the Object-oriented Approach (%)	55
4.16	Graph – Phases of the Software Development Activities that Use the Object-oriented Approach (%)	56
4.17	Graph – Programming languages Frequently used for Object-oriented (OO) Software Development (%)	58
4.18	Graph – OOAD Methodologies used for Software Development (%)	59
4.19	Graph – Object-oriented Database that are used with Software Products (%)	61
4.20	Graph – CASE Tools used to support Development Activities (%)	62
4.21	Graph – Most Important Features of Object-oriented (%)	63
4.22	Graph – Benefits of Object-oriented Technology (%)	64
4.23	Graph – Problems faced in using Object-oriented Technology (%)	66
4.24	Graph – Importance of Object-oriented (OO) Approach (%)	67
4.25	Graph – Easiness in using Object-oriented (OO) Approach (%)	68

CHAPTER 1

INTRODUCTION

1.1 Background

Object-oriented modeling and design is a new and revolutionary way of approaching and solving complex problems. This methodology is based on the concept of using models organized around real world concepts. The approach has gained great popularity among the software engineering community in recent years, and is nowadays the de-facto standard for software design in all fields. This research is aimed to study on the awareness of object-oriented approach as well as the usage of object-oriented technology in software development activities among software developers.

For this research purpose, six software development companies were identified which includes Company A, Company B, Company C, Company D, Company E, and Company F. Eighty questionnaires were distributed among these six companies and fifty-five completed questionnaires were received. This represents 69% successful respondents for this project. Based on these returned questionnaires many aspects on the usage of object-oriented approach were analyzed for this research.

Based on their responses in the questionnaires, it was discovered that there is a common company goal among these six software houses. The goal is to be a world class software house as well as to provide the best solutions and services to customers. In addition,

these companies' main businesses can be categorized as involved in software development, software consultancy, software customization, business solution providers and integrators. Furthermore it was discovered that all these companies were producing systems. The main systems produced by the companies are:

i. **Company A**

- **Corporate Information Management System (CIMS)**
- **Web Content Management System**
- **Financial Institution Statistical System (FISS)**
- **Health Management System**

ii. **Company B**

- **SMS System**
- **Project Reporting System**
- **Document Management System**

iii. **Company C**

- **Airline Ticket Payment via SMS**
- **SMS Booking System**
- **Staffware Workflow System**
- **Credit card applications**

iv. Company D

- **Ringgit Operations Monitoring System (ROMs)**
- **Electronic Payment Instructions System (e-PINTAS)**
- **Central Credit Reference (CCRIS) Batch submission**

v. Company E

- **General Insurance System (GIS)**
- **Web-based Document Management System (WEBTRA)**
- **Market Agency Support System (MASS)**
- **Broker Intelligent Decision System (BIDS)**

vi. Company F

- **Accounting and Invoicing System**
- **Warranty and Service Management System**
- **Inventory System**
- **Meter Reading Management System**

In addition to the systems that they produce, these software development companies also provide services in terms of support and maintenance (hardware and software), training, customization based on user requirements, and system integration services.

1.2 Problem Statement

Currently there are many software development houses and they use many different approaches to develop their systems or applications. Some of the approaches include structured, object-oriented and other approaches. For this research purpose, the aim is to know the status of the awareness and usage of object-oriented approach among the software development companies. This research is targeted to identify this uncertainty as there is no clear indication on the level of usage of object-oriented approach among software developers.

1.3 Objective of the project

Objectives of this project are:

- i. To study the awareness of object-oriented technology among software developers.
- ii. To identify the existing approach used among software developers to develop systems or applications in their organizations.
- iii. To analyze the level of usage of object-oriented technology in the software development activities during the system development process.

1.4 Scope of the project

- i. Covers the areas of object-oriented approach to study the awareness and usage of the approach by the software houses for developing systems or applications.
- ii. A survey will be conducted to obtain broad perspective view on the awareness and the usage of object-oriented technology among software developers. Questionnaires were distributed to software companies which targeted on the software development team members.

CHAPTER 2

LITERATURE REVIEW

The literature related to the present work was found to be highly scattered in nature. The same has been reviewed in a systematic manner as presented below.

2.1 Background of object-oriented approach

To understand the background of object-oriented approach, it is useful to note the approximate times when various object-oriented technologies were introduced.

2.1.1 Object-oriented Programming Languages

Object-oriented approaches began with the invention of object-oriented programming languages in the 1960s by Johan Dahl and Kristen Nygaard at the Norwegian Computing Center, Oslo, Norway. The first object-oriented languages were SIMULA I (1962-65) and Simula 67 (1967). Simula 67 introduced most of the key concepts of object-oriented programming which includes object and classes, subclasses (inheritance) and virtual procedures. Simula was also used as a platform by Xerox PARC for their development of Smalltalk in 1970s. In 1980s Bjarne Stroustrup started the development of C++ by bringing the key concepts of Simula into the C programming language. In the same year, the US department of Defense promoted ADA which is an object-oriented language but without inheritance [1]. Today,

Smalltalk, C++, Eiffel, Java, Ada, Visual Basic and Delphi are some of the popular object-oriented programming languages.

2.1.2 Object-oriented Methodologies

In the early 1990s approximately fifty object-oriented methodologies were proposed with a multitude of interpretations for the term 'object'. At that time, the Booch method and the Object Modeling Technique (OMT) were the ones to win the object competition [1]. An object-oriented methodology typically presents mechanisms, to model the problem in a well-defined and standardized notation. A methodology helps to create a design for the system under development and creates a common platform for the dissection and improvement of the design. Object-oriented methodologies tackle the problem by dividing the development into various stages like analysis, design and development [2]. Besides Booch and OMT other methodologies include Rapid Application Development (RAD), Fusion, Coad and Yourdon, Jacobson, Unified Modeling Language (UML), Unified Software Development Process (USDP) and many other methodologies.

2.1.3 Object-Oriented Databases

According to Zdonik and Maier (1990) the object-oriented databases were introduced in 1985. Object-oriented databases (OODB) are databases that support objects and classes. They are different from the more traditional relational database because they allow

structured sub-objects, support for methods and inheritance as well as enable each object to have its own identity, or object-id. It is also possible to provide relational operations on an object-oriented database. OODBs allow all the benefits of object-orientation, as well as the ability to have a strong equivalence with object-oriented programs, an equivalent that would be lost if an alternative were chosen, as with a purely relational database. Examples of the current commercial object-oriented DBMS (OODBMS) include GemStone, Itasca, Objectivity/DB, ObjectStore, Oritos, Poet, Jasmine, and Versant. OODBMSs can provide appropriate solutions for many types of advanced database applications [3].

2.2 Impact on introducing object-oriented approach

There are many impacts of introducing object-oriented approach in developing systems. An example of an implementation of the object-oriented approach is the Stores Controller Object Oriented Product (SCOOP) project by ESBI Computing Limited and European Software Process Improvement (EuroSPI98). The SCOOP project objective was to enable a holistic view of the impact of introducing object-oriented software development methodologies and tools [4].

The first part of the project involves the selection of the object-oriented methodology. Based on a comparative analysis and the suitability for this project, Booch was chosen as the preferred object-oriented methodology. It was primarily chosen over OMT because it deals not only with the analysis stage of a project but also the design and it also has ample documentation and support tools. The diagramming notation used in Booch is also more readily accessible and used to document almost any aspect of the system. One of the main advantages of Booch is the fact that it is extremely versatile and robust [5].

In the second part of the project, object-oriented development tools were selected. Based on the scores from questionnaires one CASE tool and two development tools were selected with the help of evaluation software as shown in Table 2.1.

Table 2.1: CASE tools and Development Environment

Category	Product	Score	Evaluation
CASE	System Architect	82.3	Very confusing interface. Rejected.
	Select CASE	79.6	GUI not intuitive but much better than System Architect. Too centered around the OMT methodology. Rejected.
	Rational Rose	72.3	Very user friendly provided good documentation and supported Booch methodology. Accepted.
Development Environment	Delphi	81.3	Excellent Visual component library – performs better than Visual Basic. Has most OO features. Accepted.
	MS Visual Basic 4.0	78.1	Shorter learning curve than Delphi but not enough OO features. Rejected.
	MS Visual J++	71.7	Immature technology. Rejected.
	MS Visual C++	83.7	All OO features, performs very well. However very poor at screen painting. Accepted.

The table shows that the selected CASE tool for this project was Rational Rose which scored 72.3% and it was accepted because it is very user friendly, provided good documentation, and supported Booch methodology. The development environment accepted was Delphi which scored 81.3% and MS Visual C++ scored 83.7%. They were chosen because of their excellent performance and have most of the object-oriented features. As a result the Booch methodology, Rational Rose, Delphi and MS Visual C++ were the accepted object-oriented approach for the SCOOP project implementation.

After the implementation of object-oriented approach in SCOOP project an assessment was made to compare the original module metrics with the SCOOP metrics to calculate which method was more productive. The following metrics were calculated as shown in Table 2.2:

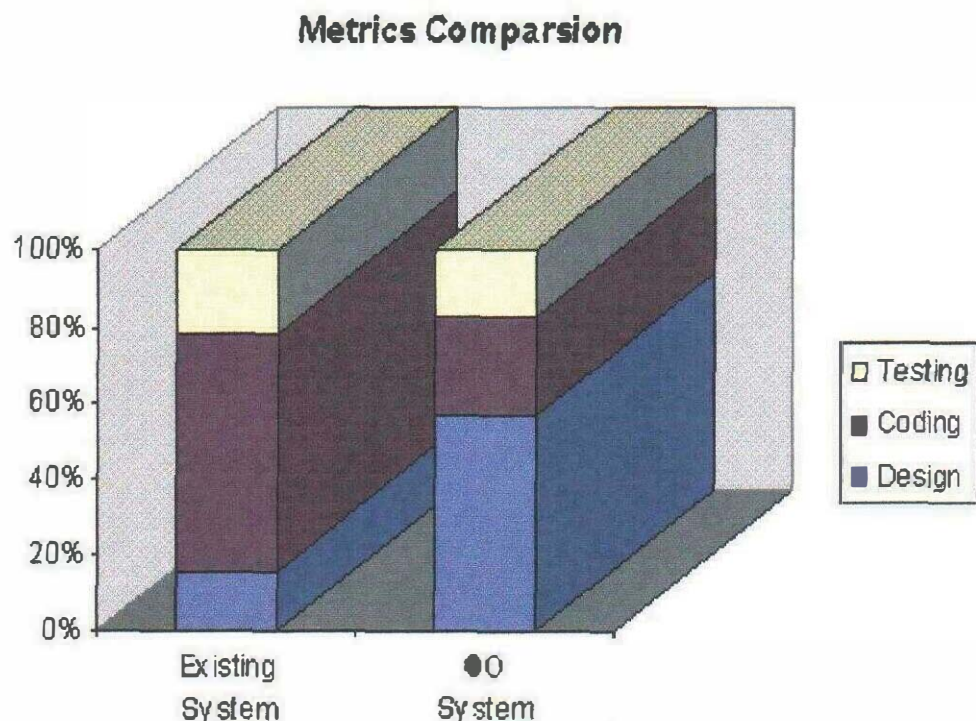


Table 2.2: Metrics Comparison between existing system and object-oriented system

Module	Design	Coding	Test	Total (days)	Lines of code
Existing System	5	20	7	32	4630
OO System	20	9	6	35	3343

Based on the metrics the following conclusions about using the object-oriented methodology were made:

- There was substantially more time spent designing in the object-oriented methodology and less time coding for the following reasons:
 - a. The design using object-oriented technique is a much more thorough process. All problems even implementation issues must be thought out at this stage. Also if a business function is left out or is added at a later stage, the class design may change radically. The designer must also have the 'big picture' view of the project and must know how the business area is used throughout the system.
 - b. There is much more documentation in the design phase. There are class diagrams, scenario diagrams, use cases and Axis of change documents. In the existing software process method there is at most two documents.
 - c. Coding takes less time due as the design documentation provides classes that can be grouped together into programmable packages. These packages can be written in isolation and accessed through interfaces.